

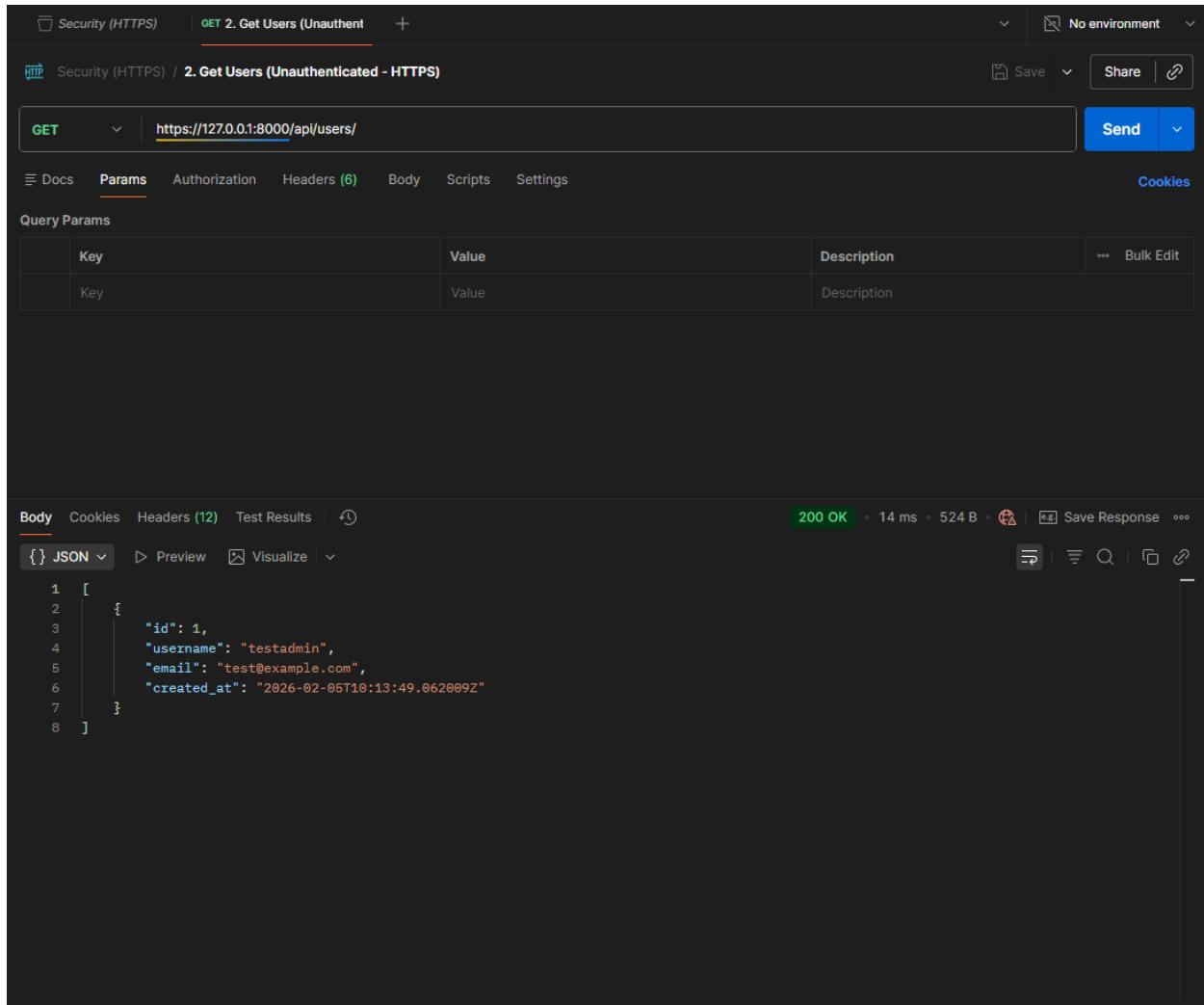
1. Get Post (Unauthenticated - HTTPS) <https://127.0.0.1:8000/api/posts/>

The screenshot shows a REST client interface with the following details:

- Title:** GET 1. Get Post (Unauthenticated)
- Method:** GET
- URL:** https://127.0.0.1:8000/api/posts/
- Headers:** Authorization, Headers (6), Body, Scripts, Settings
- Query Params:** A table with one row:

Key	Value	Description
Key	Value	Description
- Body:** JSON (empty)
- Response Status:** 200 OK
- Response Headers:** 28 ms, 421 B
- Response Content:** An empty JSON array: []

2. Get Users (Unauthenticated - HTTPS) <https://127.0.0.1:8000/api/users/>



The screenshot shows the Postman interface for a GET request to <https://127.0.0.1:8000/api/users/>. The request is successful (200 OK) with a response time of 14 ms and a size of 524 B. The response body is a JSON array containing one user object:

```
1 [  
2   {  
3     "id": 1,  
4     "username": "testadmin",  
5     "email": "test@example.com",  
6     "created_at": "2026-02-05T10:13:49.062009Z"  
7   }  
8 ]
```

3. Obtain JWT Token (HTTPS) <https://127.0.0.1:8000/api/token/>

The screenshot shows the Postman interface for a POST request to <https://127.0.0.1:8000/api/token/>. The request body is raw JSON with fields `username: "testadmin"` and `password: "admin1234!"`. The response status is 200 OK, with a response time of 63 ms and a size of 891 B. The response body is a JSON object containing `refresh` and `access` tokens.

```
1 {
2     "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJ0b2tlbl90eXBlIjoicmVmcmVzaC1sImV4cCI6MTc3MDM3NDczOCwiaWF0IjoxNzcmMjg4MzM4LCJqdGkiOiI5YWY50GZkMwYzMmM0MDNhOTg2NGM4MThiNmIwOTA2NS  
IsInVzZXJfaWQiOixin0.1N58q7-dq90y9alqH7LDcJ_40N1jm1XM1CY5jd3xUzY",  
3     "access": "eyJhbGciOiJIUzI1NiisInR5cCI6IkpXVCJ9.  
eyJ0b2tlbl90eXBlIjoiyWNnjZXNzIiwiZXhwIjoxNzcmMjkxOTM4LCJpYXQiOjE3NzAyODgzMzsImp0aSI6ImQ4MzQ1NTNjYWJmZDQzMWI4ZjNmZGVhYmMxNzY2ODY0Ii  
widXNlc19pZCIjEifQ.HIO6rvsYHB89Pcn3Lr7me3-btwtog2CYj8yliiu_y9w"  
4 }
```

4. Create User (Authenticated - HTTPS) <https://127.0.0.1:8000/api/users/>

The screenshot shows the Postman application interface. At the top, it displays "POST 4. Create User (Auther)" and "No environment". Below the header, the URL "https://127.0.0.1:8000/api/users/" is entered in the "POST" method field. The "Body" tab is selected, showing raw JSON data:

```
1 {
2     "username": "testuser1",
3     "email": "testuser1@example.com",
4     "password": "securepassword123"
5 }
```

At the bottom of the interface, the response is shown under the "Body" tab. It includes the status code "201 Created", response time "58 ms", and size "524 B". The response body is displayed as:

```
1 {
2     "id": 2,
3     "username": "testuser1",
4     "email": "testuser1@example.com",
5     "created_at": "2026-02-05T10:49:53.722266Z"
6 }
```

5. Create Post (Authenticated by Author - HTTPS) <https://127.0.0.1:8000/api/posts/>

The screenshot shows the Postman application interface. At the top, it displays "Security (HTTPS)" and "POST 5. Create Post (Author)". The URL in the header is "https://127.0.0.1:8000/api/posts/". The main area shows a POST request with the URL "https://127.0.0.1:8000/api/posts/". The "Body" tab is selected, showing the JSON payload:

```
1 {
2   "content": "My second authenticated post!"
3 }
```

Below the body, the "Test Results" section shows a successful response:

201 Created 24 ms 541 B Save Response

The response body is displayed as:

```
{ } JSON ▾
```

```
1 {
2   "id": 1,
3   "content": "My second authenticated post!",
4   "author_username": "testadmin",
5   "created_at": "2026-02-05T10:52:05.839454Z"
6 }
```

6. Update Post (Authenticated by Author - HTTPS) <https://127.0.0.1:8000/api/posts/2/>

The screenshot shows the Postman application interface. At the top, it displays the URL `PUT 6.. Update Post (Authenticated by Author) https://127.0.0.1:8000/api/posts/1/`. The main area shows a `PUT` request being sent to `https://127.0.0.1:8000/api/posts/1/`. The `Body` tab is selected, showing a JSON payload:

```
1 {  
2   "content": "Updated content for my second secure post"  
3 }
```

Below the request, the response details are shown: `200 OK`, `15 ms`, `562 B`. The response body is displayed as:

```
1 {  
2   "id": 1,  
3   "content": "Updated content for my second secure post",  
4   "author_username": "testadmin",  
5   "created_at": "2026-02-06T05:52:05.839454Z"  
6 }
```

7. Delete Post (Authenticated by Author - HTTPS) <https://127.0.0.1:8000/api/posts/2/>

The screenshot shows a REST client interface with the following details:

- Header Bar:** Security (HTTPS) | DEL 7. Delete Post (Authenticated •) | + | No environment
- Request URL:** https://127.0.0.1:8000/api/posts/1/?
- Method:** DELETE
- Body:** Raw (selected), showing the number "1".
- Response Status:** 204 No Content
- Response Headers:** 14 ms, 401 B
- Buttons:** Save, Share, Send

8. Create Regular User (Authenticated - HTTPS) <https://127.0.0.1:8000/api/users/>

The screenshot shows the Postman application interface. At the top, it displays the URL <https://127.0.0.1:8000/api/users/>. The request method is set to **POST**, and the body content type is **raw JSON**. The request body contains the following JSON payload:

```
1 {
2     "username": "regularuser",
3     "email": "regular@example.com",
4     "password": "regularpassword"
5 }
```

Below the request details, the response section is visible. The status code is **201 Created**, and the response body is:

```
1 {
2     "id": 4,
3     "username": "regularuser",
4     "email": "regular@example.com",
5     "created_at": "2026-02-05T11:03:42.922234Z"
6 }
```

9. Obtain JWT Token (Regular User - HTTPS) <https://127.0.0.1:8000/api/token/>

The screenshot shows the Postman interface with the following details:

- Request Method:** POST
- Request URL:** https://127.0.0.1:8000/api/token/
- Body Content:**

```
1 {
2     "username": "regularuser2",
3     "password": "regularpassword2"
4 }
```
- Response Status:** 200 OK
- Response Time:** 111 ms
- Response Size:** 891 B
- Response Body (JSON):**

```
1 {
2     "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
3         eyJ0b2tlbl9oeXBlijoicmVmcmVzaC1sImV4cI6Mtc3MDM3NjIxMSwiaWF0IjoxNzcwMjg50DExLCJqdGkiOiI50ThhODAwNmJkMzY0YTvhOGNhMzI40TQ4Yzc4ZmM0Yy
4             IsInvzXJfawQioii1In0.I1jQ4ML2WBTL2kxe7PXFtWHzdbiU3k8rTqhKzJBMs8",
5     "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
6         eyJ0b2tlbl9oeXBlijoicmVmcmVzaC1sImV4cI6Mtc3MDM3NjIxMSwiaWF0IjoxNzcwMjg50DExLCJqdGkiOiI50ThhODAwNmJkMzY0YTvhOGNhMzI40TQ4Yzc4ZmM0Yy
7             widXNlcl9pZCI6IjUiifQ.Jn6ysrw0TstbzDjvpkU-6w7vwewm2ktPIF9tjuxfPvA"
```

10. Regular User Create Post (HTTPS) <https://127.0.0.1:8000/api/posts/>

The screenshot shows the Postman application interface. At the top, there's a header bar with tabs for 'Security (HTTPS)', 'POST 10. Regular User Creat •', and '+'. Below the header, the URL is set to 'https://127.0.0.1:8000/api/posts/'. The method is set to 'POST'. On the right side of the header, there are 'Save' and 'Share' buttons, and a status indicator 'No environment'.

The main workspace has tabs for 'Body', 'Params', 'Authorization', 'Headers (9)', 'Body' (selected), 'Scripts', and 'Settings'. Under 'Body', the 'raw' tab is selected, showing the JSON payload:

```
1 {  
2   "content": "Post by regular user 2"  
3 }
```

On the right side of the workspace, there are 'Cookies', 'Schema', and 'Beautify' buttons. Below the workspace, there are tabs for 'Body', 'Cookies', 'Headers (12)', and 'Test Results'. The 'Test Results' tab is selected, showing a green success message: '201 Created' with a timestamp of '27 ms' and a size of '537 B'. There are also 'Save Response' and other UI elements like copy and search buttons.

11. Superuser Update Regular User Post (RBAC Fails - HTTPS)

<https://127.0.0.1:8000/api/posts/2/>

The screenshot shows a POST request in Postman. The URL is <https://127.0.0.1:8000/api/posts/2/>. The request method is PUT. The body contains the following JSON:

```
1  {
2    "content": "Superuser trying to change regular user's post"
3 }
```

The response status is 403 Forbidden. The response body is:

```
1  {
2    "detail": "You do not have permission to perform this action."
3 }
```

12. Regular User Update Own Post (RBAC Succeeds - HTTPS)

<https://127.0.0.1:8000/api/posts/>

The screenshot shows a POST request in the Postman application. The URL is `https://127.0.0.1:8000/api/posts/2/`. The request method is `PUT`. The body contains the following JSON:

```
1 {
2     "content": "Regular user updates their own post"
3 }
```

The response status is `200 OK`, with a response time of 23 ms and a response size of 559 B. The response body is:

```
1 {
2     "id": 2,
3     "content": "Regular user updates their own post",
4     "author_username": "regularuser2",
5     "created_at": "2026-02-05T11:14:08.381411Z"
6 }
```

13. Create Post (Unauthenticated Attempt - HTTPS) <https://127.0.0.1:8000/api/posts/>

The screenshot shows a POST request to <https://127.0.0.1:8000/api/posts/>. The request body contains a JSON object with a single key "content": "Attempt to post without a token.". The response status is 401 Unauthorized, with a response body indicating "Authentication credentials were not provided."

Request Body:

```
1 {  
2   "content": "Attempt to post without a token."  
3 }
```

Response Headers:

```
401 Unauthorized
```

Response Body:

```
1 {  
2   "detail": "Authentication credentials were not provided."  
3 }
```